Amazon Fine Food Review:

# Conclusion:

## Step 1:

After performing **de-duplication** of reviews, we were left with **~364k reviews**, this is considered as **large data**, as every review had on **average ~96 words**, so in total there would be roughly around **~34944k words (huge)**

So, in order to get the best model which gives the optimal result, we tried building (training) the model on a sample of full data, i.e., we took a **random sample of 5% from full data** and build various models on this sample.

Featurization Technique used for all the below mentioned models is **Word2Vec with Tf-Idf**, as it captures the essence of both word2vec and Tf-Idf.

Results, of these models are shown below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class Imbalance Handled | | | | | | | |
| Model | Recall | | Precision | | F1 Score | | Accuracy |
| Negative | Positive | Negative | Positive | Negative | Positive |
| Logistic Regression | 0.81 | 0.78 | 0.42 | 0.96 | 0.55 | 0.86 | 0.79 |
| Random Forest | 0.77 | 0.75 | 0.38 | 0.94 | 0.5 | 0.84 | 0.76 |
| Support Vector Machine | 0.82 | 0.79 | 0.44 | 0.96 | 0.57 | 0.87 | 0.8 |
| Naive Bayes | 0.76 | 0.74 | 0.36 | 0.94 | 0.49 | 0.83 | 0.74 |
| Neural Network | 0.83 | 0.77 | 0.41 | 0.96 | 0.55 | 0.85 | 0.78 |
| K-NN | 0.76 | 0.7 | 0.33 | 0.94 | 0.46 | 0.8 | 0.71 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class Imbalance Not handled | | | | | | | |
| Model | Recall | | Precision | | F1 Score | | Accuracy |
| Negative | Positive | Negative | Positive | Negative | Positive |
| Logistic Regression | 0.44 | 0.96 | 0.71 | 0.9 | 0.54 | 0.93 | 0.88 |
| Random Forest | 0.05 | 1 | 0.84 | 0.85 | 0.1 | 0.92 | 0.85 |
| Support Vector Machine | 0.39 | 0.97 | 0.75 | 0.89 | 0.51 | 0.93 | 0.88 |
| Naive Bayes | 0.56 | 0.9 | 0.52 | 0.91 | 0.54 | 0.91 | 0.84 |
| Neural Network | 0.51 | 0.95 | 0.68 | 0.91 | 0.58 | 0.93 | 0.88 |
| K-NN | 0.31 | 0.97 | 0.68 | 0.88 | 0.42 | 0.92 | 0.86 |

The results shown are for supervised models, unsupervised models were also considered, but they were performing very badly as accuracy was in range of ~50% which is not acceptable, thus, we stuck with supervised models.

From, the table above it can be concluded that **Logistic Regression, Support Vector Machine and Neural Network were seen to perform the best, giving 88% accuracy**.

Also, when the size of dataset was increased, performance of SVM was found to degrade when compared with Logistic Regression, and Neural Network required comparatively much larger computation time, so best model for this dataset was concluded to be Logistic Regression.

Also, when dealt with class imbalance problem the accuracy was seen to degrade, based on this we draw a conclusion that it not a wise option to take this issue into consideration.

## Step 2:

Once, we finalized the best model for the given dataset using a random sample, i.e., Logistic Regression, next step is to get the best feature extraction technique for the given dataset.

For this purpose, we used 4 different feature extraction technique namely,

* Bag of Words (BOW) (n-grams) **(Ashish n-grams konsa use kiya hai fill kar dena)**
* Tf-Idf (Term frequency – Inverse Document Frequency) (n-grams)
* Word2Vec with Average
* Word2Vec with Tf-Idf

We build feature extraction matrix for each of BOW, Tf-Idf, W2V+Avg, W2V+Tf-Idf and then applied Logistic Regression on this matrix and computed the results, which are shown below, the results shown are for full dataset, train-test ratio 90:10.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Splitting into train and test done after forming BOW, TFIDF, W2V on full data | | | | | | | |
| Binary Classification+Full Dataset+Logistic+Randomsplitting | | | | | | | |
| Featurization Technique | Precision | | Recall | | F1 Score | | Accuracy |
| Neg | Pos | Neg | Pos | Neg | Pos |
| BOW | 0.62 | 0.97 | 0.85 | 0.9 | 0.71 | 0.93 | 0.89 |
| TFIDF | 0.6 | 0.97 | 0.88 | 0.89 | 0.71 | 0.93 | 0.89 |
| W2V+Avg | 0.48 | 0.97 | 0.85 | 0.83 | 0.61 | 0.89 | 0.83 |
| W2V+TFIDF | 0.44 | 0.96 | 0.82 | 0.8 | 0.57 | 0.88 | 0.81 |

The table explains, that best results are obtained with Tf-Idf and BOW, but when the model was running, multiple times with random splitting in train and test, Tf-Idf was found to perform better than BOW.

Based on this observation we can conclude that the best feature extraction technique for this dataset is Tf-Idf.

Also, Logistic regression model with multiclass classification – 3 class and 5 class were trained and tested with BOWs, and Tf-Idf, which yield following results,

|  |  |
| --- | --- |
| 3 class+fulldata+Logistic | |
| Featurization Technique | Accuracy |
|  |
| BOW | 0.7709 |  |
| TFIDF | 0.7513 |  |
| W2V+Avg | 0.6925 |  |
| W2V+TFIDF | 0.5 |  |

Almost ~10% accuracy is decreased compared to binary classification in each case,

|  |  |
| --- | --- |
| 5 class +100000+Logistic | |
| Featurization Technique | Accuracy |
| BOW | 0.6169 |
| TD-IDF | 0.577 |

Also, for 3 class, text preprocessing technique – Lemmatization was used instead of Stemming which yielded slightly better results – 0.8046 accuracy with BOW and Logistic Regression Model but, Lemmatization required a significant amount of computation time, when compared to stemming and thus was not found significant enough, so all the models were build using stemming. **(Vivek isko acche se edit kar dena)**

Based on the results for multiclass classification shown in the tables above, we observed that accuracy degrades as we increase number of classes of classification, based on this observation, we came to a conclusion that binary classification is best suited for this dataset.

# Result:

Examining all the possible models, supervised and unsupervised, all feature extraction techniques, all text preprocessing techniques, different number of class classification, the best model with features was found to be,

* Model – Logistic Regression (Neural Network can be tuned if high computation power is available)
* Feature Extraction Technique – Tf-Idf (BOW can also be used)
* Number of Classes – Binary (2)
* Text Preprocessing – Stemming (Lemmatization, if high computation power is available)

Results of best model with above mentioned features are as below:

Accuracy: abc **(ashish isko fill kar dena)**

Precision: abd

Recall: abd

F1-Score: abs